Package ‘MclustSepCov’

January 2, 2019

Type Package

Title Model based Clustering via Mixture Distribution under Covariance Separability

Version 1.0

Date 2019-01-02

Author Seongoh Park [aut, cre], Johan Lim [aut], Hye Jeong Choi [aut], Minjung Kwak [aut]

Maintainer Seongoh Park <seongohpark6@gmail.com>

Description Perform clustering with multivariate longitudinal data based on the Gaussian mixture distribution with separable covarinace matrix.

License GPL-3

Encoding UTF-8

Imports Rcpp (>= 0.12.19)

Suggest mvtnorm

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 6.1.1

NeedsCompilation yes

R topics documented:

`MclustSepCov-package` *Model based Clustering via Mixture Distribution under Covariance Separability*

Description

Perform clustering with multivariate longitudinal data based on the Gaussian mixture distribution with separable covarinace matrix.

Details

The DESCRIPTION file:

Package:	MclustSepCov
Type:	Package
Title:	Model based Clustering via Mixture Distribution under Covariance Separability
Version:	1.0
Date:	2019-01-02
Author:	Seongoh Park [aut, cre], Johan Lim [aut], Hye Jeong Choi [aut], Minjung Kwak [aut]
Maintainer:	Seongoh Park <seongohpark6@gmail.com>
Description:	Perform clustering with multivariate longitudinal data based on the Gaussian mixture distribution with se
License:	GPL-3
Encoding:	UTF-8
Imports:	Rcpp (>= 0.12.19)
Suggest:	mvtnorm
LinkingTo:	Rcpp, RcppArmadillo
RoxygenNote:	6.1.1

Index of help topics:

```
MclustSepCov-package    Model based Clustering via Mixture Distribution
                        under Covariance Separability
Mclust_SEP_cpp          The model-based clustering for longitudinal
                        data
Mclust_SEP_each_cpp     The maximum likelihood estimation of the
                        mixture distribution
Optimization            Newton-Raphson's algorithm to find the optimal
                        temporal correlation
getCovariance           Generate temporal covariance matrices
```

Author(s)

Seongoh Park [aut, cre], Johan Lim [aut], Hye Jeong Choi [aut], Minjung Kwak [aut]

Maintainer: Seongoh Park <seongohpark6@gmail.com>

---

getCovariance *Generate temporal covariance matrices*

Description

Return a covariance matrix with temporal structure.

Usage

```
getCovariance(q, rho, type)
```

Arguments

q	Dimension of a covariance matrix.
rho	Temporal correlation in $(-1, 1)$. For type='CS', $-1/\sqrt{q-1}$ is a lower bound of rho for the returned matrix to be positive definite.
type	A character string indicating one of types of temporal structure; autoregressive model if type='AR' and compound symmetry model if type='CS'. See 'Details' for their structures.

Details

Following temporal structures are available.

1. Autogressive structure: $V = \left(\rho^{|i-j|}; 1 \le i, j \le q\right)$.
2. Compound symmetry structure: $V = \left(\mathrm{I}(i = j) + \rho\mathrm{I}(i \ne j); 1 \le i, j \le q\right)$.

Value

A q-by-q temporal covariance matrix with diagonals 1.

Examples

```
getCovariance(3, 0.3, "AR") # AR
getCovariance(3, 0.3, "CS") # CS

# AR structure with heterogeneous variances
hvar <- c(1, 2, 3) # variances
diag(sqrt(hvar)) %*% getCovariance(3, 0.3, "AR") %*% diag(sqrt(hvar))
```

Mclust_SEP_cpp *The model-based clustering for longitudinal data*

Description

This is a wrapper function of Mclust_SEP_each_cpp. All arguments except save_fit will be passed to Mclust_SEP_each_cpp.

Usage

```
Mclust_SEP_cpp(Y, p, q, Ks, type_cov, tol = 0.001, maxit = 500,
  save_fit = TRUE)
```

Arguments

Y	A r-by-(p*q) matrix where r is the sample size, and ordering of columns should be carefully set (see 'Details').
p, q	An integer value for the number of multi-variables and the number of time points, respectively.
Ks	A sequence of positive integers indicating the number of mixture components, each of which will be used in K of Mclust_SEP_each_cpp.

type_cov	A sequence of character strings indicating covariance structures, each of which will be used in Mclust_SEP_each_cpp. Default is 'all', which runs all available models. See 'Details'.
tol	Tolerance constant for convergence. Default is 1e-3.
maxit	Maximum number of iterations. Default is 500.
save_fit	A logical value indicating whether to save all fitted mixture models. If FALSE, the best model is only available by best_model. Default is TRUE.

Details

The first q components from each row of Y denote q variables at time point 1, the second q are those at time point 2, and so on until time point p. Under separability, the covariance matrix of row vectors of Y is represented by $U_{p\times p} \otimes V_{q\times q}$ for some covariance factors $U_{p\times p}, V_{q\times q}$.

type_cov should be in "XXX-YYY" format. "XXX" is for the multivariable covariance $U_{p\times p}$, and "YYY" for the temporal covariance $V_{q\times q}$. They will be passed respectively to type_vari and type_time in Mclust_SEP_each_cpp. Available options are as follows;

- Heteroscadatsic covariance structure : VVV-VUN (unstructured), VVV-VAR (AR), VVV-VCS (CS).
- Homoscadatsic covariance structure : EEE-EUN (unstructured), EEE-EAR (AR), EEE-ECS (CS).

For initialization of cluster membership, see 'Details' in Mclust_SEP_each_cpp.

Value

A list with components:

best_model	A list of the mixture models with the largest BIC. If there is a tie, they are all returned.
bic_table	Table filled with BIC values. Type of covariance models are given by rows and values in Ks by columns.
res_Mclust_SEP	If save_fit is TRUE, all fitted models are stored. It is a nested list with the first layer corresponding to covariance models specified in type_cov and the second to values of Ks.

See Also

Mclust_SEP_each_cpp

Examples

```
# Gaussian mixture model with two components
K <- 2
p <- 2
q <- 3
U <- lapply(1:K, function(noarg) getCovariance(p, 0.3, "AR"))
V <- lapply(1:K, function(noarg) getCovariance(q, 0.2, "CS"))
Sigma <- Map(kronecker, U, V) # separable covariance matrix
mu <- list(rep(0, p * q), 5 / sqrt(p*q) * rep(1, p * q)) # distinct mean vectors
Y <- vector(mode = "list", length = K)
for(i in 1:K){
  Y[[i]] <- mvtnorm::rmvnorm(n = 20, mean = mu[[i]], sigma = Sigma[[i]])
}
fit <- Mclust_SEP_cpp(Y = Reduce(rbind, Y), p = p, q = q, Ks = 2, type_cov = "EEE-ECS")
```

Mclust_SEP_each_cpp — *The maximum likelihood estimation of the mixture distribution*

Description

Perform the EM algorithm for fitting the finite Gaussian mixture distribution with covariance separability.

Usage

```
Mclust_SEP_each_cpp(Y, p, q, K, type_vari, type_time, tol = 0.001,
  maxit = 500L)
```

Arguments

Y	Same as in Mclust_SEP_cpp.
p, q	Same as in Mclust_SEP_cpp.
K	A positive integer indicating the number of mixture components.
type_vari, type_time	A character string indicating a structure of covariance factors, passed from Mclust_SEP_cpp. See 'Details' for available options.
tol	Same as in Mclust_SEP_cpp.
maxit	Same as in Mclust_SEP_cpp.

Details

Cluster membership from 1 to K is randomly assigned to each sample.

type_vari specifies a type of the multivariable covariance $U_{p \times p}$;

- Heteroscadatsic : VVV (unstructured)
- Homoscadatsic : EEE (unstructured)

and type_time the temporal covariance $V_{q \times q}$;

- Heteroscadatsic : VUN (unstructured), VAR (AR), VCS (CS)
- Homoscadatsic : EUN (unstructured), EAR (AR), ECS (CS)

Value

A list with components:

loglik	The log-likelihood function.
df	The degrees of freedom or the number of parameters of the mixture model.
BIC	The Bayesian information crieteria.
K	K from the input.
id_cluster	Cluster membership of samples.
wt_cluster	A matrix of dimension r-by-K whose row represents the maximum a posteriori of a sample.

EM_iter	The number of iterations in the EM algorithm.
mu	A matrix of the estimated mean whose row is the mean vector of a mixture component.
U,V	A cube containing K slices of the estimated covariance matrix.
type_vari, type_time	type_vari, type_time from the input.

See Also

Mclust_SEP_cpp

Examples

```
# Gaussian mixture model with two components
K <- 2
p <- 2
q <- 3
U <- lapply(1:K, function(noarg) getCovariance(p, 0.3, "AR"))
V <- lapply(1:K, function(noarg) getCovariance(q, 0.2, "CS"))
Sigma <- Map(kronecker, U, V) # separable covariance matrix
mu <- list(rep(0, p * q), 5 / sqrt(p*q) * rep(1, p * q)) # distinct mean vectors
Y <- vector(mode = "list", length = K)
for(i in 1:K){
  Y[[i]] <- mvtnorm::rmvnorm(n = 20, mean = mu[[i]], sigma = Sigma[[i]])
}
fit <- Mclust_SEP_each_cpp(Y = Reduce(rbind, Y), p = p, q = q, K = 2, type_vari = "EEE", type_time = "ECS")
```

Optimization — *Newton-Raphson's algorithm to find the optimal temporal correlation*

Description

Solve the constrained minimization problem using the log-barrier method to find the maximum likelihood estimator (MLE) of temporal correlation. The objective function is described in 'Details'.

Usage

```
LB_algorithm_cpp(a, Z, type, rho0, lambda = 1, maxit = 500L)
```

Arguments

a	A positive constant. See 'Details'.
Z	A matrix of sample vectors at row.
type	A character string indicating a type of temporal covariance matrix. Available options are 'AR' and 'CS'.
rho0	An initial value for the temporal correlation coefficient. We empirically found that 0.001 works well for type='AR' and 0.5 for type='CS'.
lambda	A positive constant multiplied to the log-barrier term. Default is 1.
maxit	The maximum number for iterations. Default is 500.

Details

The objective function is divided into two parts; the Gaussian log-likelihood function (up to constant multiplication) with mean 0 and covariance matrix $\Sigma = \Sigma(\rho)$ and the log-barrier function. The former is written by

$$h(\rho; a, Z) = a \log|\Sigma| + \mathrm{tr}(\Sigma^{-1} S),$$

where $a > 0$ and $S = Z^{\mathrm{T}} Z$, and the latter is

$$b(\rho; u, l) = \log(u - \rho) + \log(\rho - l),$$

where u, l is an upper and a lower bound of ρ, respectively. These quantities depend on `type` as follows;

- If `type='AR'`, $\Sigma = \left(\rho^{|i-j|}; 1 \le i, j \le q\right)$ and $l = -1, u = 1$,
- if `type='CS'`, $\Sigma = \left(\mathrm{I}(i = j) + \rho \mathrm{I}(i \neq j); 1 \le i, j \le q\right)$ and $l = -1/\sqrt{q-1}, u = 1$,

where $q =$`ncol(Z)`. The objective function is, hence,

$$h(\rho; a, Z) - \lambda\, b(\rho; u, l).$$

Examples

```
q <- 10
# AR model
set.seed(6)
Y <- mvtnorm::rmvnorm(100, rep(0, q), getCovariance(q, 0.3, "AR"))
LB_algorithm_cpp(a = nrow(Y), Z = Y, rho0 = 1e-3, type = "AR")

# CS model
set.seed(6)
Y <- mvtnorm::rmvnorm(100, rep(0, q), getCovariance(q, 0.3, "CS"))
LB_algorithm_cpp(a = nrow(Y), Z = Y, rho0 = 1e-3, type = "CS")
```

Index